IAP12 Rec'd PCT/PTO 03 FEB 2006

ARRANGEMENT AND METHOD FOR CONNECTING A NODE IN A DISTRIBUTED SYSTEM

Field of the Invention

This invention relates to distributed systems and particularly (thoughy not exclusively) to hard real-time systems using static TDMA (Time Division Multiple Access) based medium arbitration that should remain operational even if subjected to the arbitrary failure of a single processing node.

Background of the Invention

In the field of this invention it is known that in a distributed processing system having a plurality of nodes, while a processing node can fail in an arbitrary way, it is necessary to assure that a single faulty fail-uncontrolled processing node does not disrupt communication among fault-free processing nodes. In order to achieve this objective it is known to use either

(a) a fully connected network topology, or

(b) a multi-drop transmission line topology, or

(c) a star topology containing an intelligent central distribution unit, or

(d) a multi-access topology with an 'anti-jabbering' unit in the form of a bus guardian at the outgoing network interface of each processing node.

However, these approaches have the disadvantages that:

(a) a fully connected network topology involves high cost, and an unfeasible network structure;

(b) a multi-drop transmission line topology requires a receiver for every multi-drop transmission channel;

(c) a star topology containing an intelligent central distribution unit requires high complexity in the distribution unit, increasing susceptibility to faults;

(d) in a multi-access topology with an 'anti-jabbering' unit in the form of a bus guardian at the outgoing network interface of each processing node, the bus guardian is susceptible to spatial proximity faults, and potential functional

dependency between the bus guardian and the communication unit within the processing node.

A need therefore exists for a scheme for interconnecting processing nodes in a distributed system wherein the abovementioned disadvantages may be alleviated.

Statement of Invention

In accordance with a first aspect of the present invention there is provided an arrangement for connecting a node in a distributed system as claimed in claim 1.

In accordance with a second aspect of the present invention there is provided a method of operating a node in a distributed system as claimed in claim 11.

Brief Description of the Drawings

Various methods and arrangements for interconnecting fail-uncontrolled processors in a dependable distributed system incorporating the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 shows a block schematic illustration of a known bus guardian based distributed processing system;

FIG. 2 shows a block schematic illustration of a node guardian arrangement which may be used in the present invention;

FIG. 3 shows a block schematic illustration of a distributed processing system developed from the bus guardian based system of FIG. 1 and utilising the node guardian arrangement of FIG. 2;

FIG. 4 shows a block schematic illustration of an improved version of the distributed processing system of FIG. 3 for making use of dual channel capabilities; and

FIG. 5 shows a block schematic illustration of a distributed processing system containing four central nodes, demonstrating how the node guardian approach of FIG. 4 scales to a larger system.

Description of Preferred Embodiments

Referring firstly to FIG. 1, a known bus guardian based distributed processing TTP/C™ (Time Triggered Protocol class C) network system 100 includes nodes 110, 120, 130, 140, 150, 160, 170, 180; each of the nodes has a bus guardian BG 111, 121, 131, 141, 151, 161, 171, 181. The nodes 110, 120 and 130 are coupled via a common channel 190 to the nodes 140 and 150, and constitute an error containment region 101. The nodes 160, 170 and 180 are coupled via a common channel 195 to the nodes 140 and 150, and constitute an error containment region 102. Babbling idiot faults occurring in nodes 110, 120 or 130 act on 140 and 150 (depicted by fault propagation path a) but not on nodes 160, 170 and 180, while faults occurring in nodes 160, 170 or 180 act on 140 and 150 (depicted by fault propagation path b) but not on nodes 110, 120 and 130. Faults in 140 or 150 (fault propagation path x1 and x2), however, act on both error containment regions, i.e., nodes 110, 120, 130, 160, 170 and 180 including nodes 140 and 150.

The invention allows the problem of protecting a computational node in a distributed system against 'babbling idiot' failures (when a faulty node continually broadcasts a message, which takes over the bus) to be solved by equipping each node with a 'node guardian'. The structure of such a node guardian is shown in FIG. 2.

The node guardian 200 consists of a set of switches (FIG. 2 shows an example of three, namely: 240, 241, 242) that connect input signals from a set of bus drivers 230, 231, 232 to a receiver 271 of the communication processor 280 via a logical-OR operation 260 and a control unit 250 that interoperates with the communication processor 280 via a control unit 272 and controls the state of each respective switch 240, 241, 242. It will be appreciated that input switches 240, 241, 242 combined with a logic element 260 act as an in input multiplexer under the control of the control unit 250.

It will be understood that control units 272 and 250 are only separated to demonstrate conformance to the known FlexRay™ architecture, and may otherwise be commonly provided.

The node guardian 200 protects the receiver of the communication processor against jabbering nodes by enabling and disabling the respective switches according to a static TDMA schedule (which will be understood and need not be described in further detail). The node guardian 200 implements an input protection boundary, which means it may share the same clock, power supply, etc., with the communication processor 280. Another key difference compared to the bus guardian approach is that with the node guardian 280 protection occurs outside of the sphere of the faulty node and within the sphere of the fault-free device.

FIG. 3 shows an example of a hierarchical distributed processing network system system developed from the purely bus guardian based system of FIG. 1 and utilising the bus guardian arrangement 200 of FIG. 2. In the system 300 of FIG. 3, nodes 310, 320, 330, 360, 370 and 380 implement the bus guardian approach as in the system of FIG. 1, while two nodes 340 and 350 each incorporate the arrangement 200 of FIG. 2 in 341 and 351 and are based on the node guardian approach. The nodes 310, 320 and 330 are coupled via a common channel 390 to the node 340 and constitute error containment region 301, and are also coupled via the channel 390 to the node 350. The nodes 360, 370 and 380 are coupled via a common channel 395 to the node 350 and constitute an error containment region 302, and are also coupled via the channel 395 to the node 340. The node 340 is coupled to the node 350 via path 393, and the node 350 is coupled to the node 340 via path 398. Path 393 enables node 340 to communicate with node 350 even if a jabbering fault in node 310, 320 or 330 has penetrated past the respective bus guardian 311, 312 or 313 and blocked channel 390. Path 398 serves in a corresponding way for node 350. In the system of FIG. 3, a fault occurring in error containment region 301 (i.e., nodes 310, 320, 330 or 340) is confined to region 301 (fault propagation path a and x1) and cannot impact the nodes in error containment region 302. The same holds true *vice versa* for faults originating in error containment region 302. Hence, a clear concept of confinement is implemented.

FIG. 4 shows an improved version of the example shown in FIG. 3 that makes use of the dual channel capabilities provided in particular by FlexRay™. Additionally to the system of

FIG. 3, in the system 400 of FIG. 4 the nodes 410, 420 and 430 are coupled to the nodes 440 and 450 via a common channel 491, and the nodes 460, 470 and 480 are coupled to the nodes 440 and 450 via a common channel 495; the node 440 is coupled to the node 450 via path 492, and the node 450 is coupled to the node 440 via path 497. In the system of FIG. 4, it is possible to tolerate transient and one permanent channel failure in either fault containment region. In case of a failure of channel 490, for example, the nodes 410, 420, 430 and 440 can still communicate via channel 491. The same holds true *vice versa* for a channel failure in fault containment region 402.

FIG. 5 shows how the node guardian approach (as shown, for example, in FIG. 3) scales to a large system containing four central nodes 540, 545, 550, 555 each having a node guard arrangement 200 for each channel (the figure only being completed for one channel). In the system 500 of FIG. 5, as can be seen by comparison with FIG. 3, two further central nodes 545 and 555 have been added to the nodes 540 and 550, the nodes 545 and 555 being provided in error containment regions 501 and 502 respectively, the nodes 510, 520, 530 being coupled directly to central nodes 540, 545 and 550, the nodes 560, 570, 580 being coupled directly to central nodes 540, 545 and 555, and the nodes 540, 545, 550, 555 being cross-coupled. The cross-coupling enables nodes 540, 545, 550 and 555 to maintain communication even if, for example, the channel connecting 510, 520 and 530 with 540, 545 and 550 fails.

It will be appreciated that a key benefit of the node guardian approach compared to the bus guardian approach is that it is not necessary to define an error containment boundary within the transmitting node. This eliminates many problems encountered with the bus guardian approach concerning avoiding common failure modes, such as independent clock sourcing, independent power supply, testing and test interaction.

It will be understood that the method and arrangement for interconnecting fail-uncontrolled processors in a dependable distributed system described above provides the following advantages:

The invention transfers the problem of fault containment from the output interface of a potentially faulty processing node to the input interface of fault-free processing nodes. By doing so, problems encountered by spatial proximity faults or functional dependencies within

a faulty processing node that may jeopardize fault containment at its output interface are mitigated.